Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Кафедра програмної інженерії

КУРСОВА РОБОТА

ПОЯСНЮВАЛЬНА ЗАПИСКА

з дисципліні “Бази даних”

«Інформаційна система «Магазин цифрових компакт-дисків»»

Керівник, доц. каф ПІ Мазурова О.О.

Студент гр. ПІ-15-3 Крилов К.Ю.

Комісія: Мазурова О.О.

Зибіна К.В.

Черепанова Ю.Ю.

Харків 2016\_\_\_\_\_\_Харківський національний університет радіоелектроніки\_\_\_\_\_\_\_\_\_\_

**Кафедра** \_\_\_Програмної інженерії \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Дисципліна**\_\_\_Базиданих\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Спеціальність**\_Програмнаінженерія\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Курс** \_\_\_2\_\_\_\_\_\_\_\_\_\_\_

**Група**\_\_\_ПІ-15-3\_\_\_\_\_\_\_\_\_**Семестр**\_\_\_\_4\_\_\_\_\_\_\_\_\_\_\_

**ЗАВДАННЯ**

**на курсову роботу студента**

\_\_\_\_Безсмертного Олександра Петровича \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**1. Тема роботи:** \_\_\_\_«Інтернет-магазин електроніки»\_\_\_\_\_\_\_\_\_\_\_\_

**2. Строк здачі закінченої роботи** \_\_\_28.12.2016\_\_\_\_\_\_\_\_

**3. Вихідні дані для роботи:** методичні вказівки до виконання курсової роботи,\_вимоги до інформаційної системи, предметна область, що пов’язана з\_діяльністтю санаторія\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**4. Зміст розрахунково - пояснювальної записки:** вступ, аналіз предметної області; постановка задачі; проектування бази даних; опис програми; висновки; перелік посилань.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**5. Перелік графічного матеріалу:** загальна схема концептуальної моделі, ER-діаграма, структура 1НФ, 2НФ, 3НФ, схема БД в 3НФ, UML-діаграми, копії екранів (“скріншоти”) прикладної програми, приклади звітів прикладної програми\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**6. Дата видачі завдання** \_\_\_09.09.16 р.\_\_\_

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |  |
| --- | --- | --- | --- |
| Номер | Назва етапів курсової роботи | Строк виконання етапів роботи | Примітки |
| 1 | Аналіз предметної області | 9.09.16 – 24.09.16 | Виконано |
| 2 | Постановка задачі | 20.09.16 – 30.09.16 | Виконано |
| 3 | Побудова ER-діаграми бази даних | 27.09.16 – 15.10.16 | Виконано |
| 4 | Оформлення розділів 1, 2 та 3.1, 3.2 пояснювальної записки | 15.10.16 - 27.10.16 | Виконано |
| 5 | Перша контрольна точка з курсового проекту | 24.10.16 – 28.10.16 | Виконано |
| 6 | Нормалізація бази даних | 20.10.16 - 10.11.16 | Виконано |
| 7 | Створення демо-версії програми | 20.10.16 – 20.11.16 | Виконано |
| 8 | Тестування програми, наповнення бази даних | 15.11.16 - 25.11.16 | Виконано |
| 9 | Друга контрольна точка з курсового проекту | 21.12.16– 02.12.16 | Виконано |
| 10 | Реалізація остаточної версії програми | 1.12.16-15.12.16 | Виконано |
| 11 | Оформлення інших розділів пояснювальної записки | 1.11.16 – 15.12.16 | Виконано |
| 12 | Захист курсового проекту (третя контрольна точка) | 12.12.16- 23.12.16 | Виконано |

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Безсмертний О.П.

Керівник \_\_\_\_ доц. Мазурова О.О.. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2016 р.

РЕФЕРАТ

Пояснювальна записка: 32с., 33 рис., 3 джерела.

Метою роботи є розробка інформаційної системи для магазину цифрових компакт-дисків.

Метод розробки ­– концептуальне, ER- і UML- моделювання предметної області, об’єктно-орієнтований підхід до розробки програмного додатку.

Результати: спроектували базу даних і створили інформаційну систему для роботи з даними.

БАЗА ДАНИХ, ВІДНОШЕННЯ, СУБД, ІНФОРМАЦІЙНА СИСТЕМА, ЦИФРОВИЙ КОМПАКТ-ДИСК, PYTHON, SQL SERVER.

СОДЕРЖАНИЕ

Введение……………………………………………………………………. …….6

1 Анализ и концептуальное моделирование предметной области...……..…....7

1.1 Анализ предметной области……………………………………. …….. 7

1.2 Концептуальное моделирование……………………………………. 8

2 Постановка задачи………………………………………………………… 12

3 Проектирование базы данных………….…………………………………… 14

3.1 UML-моделирование……………………………………………………. 14

3.2 Построение ER-диаграммы……………………………………........... 14

3.3 Построение базы данных в третьей нормальной форме………………. 16

4 Описание программы…………………………………………………………20

4.1 Общие сведения……………………………………………………………..20

4.2 Вызов и загрузка…………………………………………………………….20

4.3 Назначение и логическая структура……………………………………….20

4.4 Описание физической модели базы данных …………………………….. 21

4.5 Описание программной реализации……………………………………….24

Выводы…………………………………………………………………….…….34

Перечень ссылок…….………………………......................................................35

ВВЕДЕНИЕ

Прогресс неумолимо движется вперед, заставляя человека и все аспекты его жизни постоянно модернизироваться, чтобы «поспевать в ногу» с ним. Ежесекундное совершенствование, разбиение сложных конструкций до уровня мелких подзадач помогают избежать распространенных ошибок во время анализа и моделирования предметной области.

Разрабатываемая информационная система «Интернет-магазин электроники» предназначена для торговли электроникой в интернете. Учитывая специфику данной области, основными целями являются реализация удобного пользовательского интерфейса и возможности учета товарооборота.

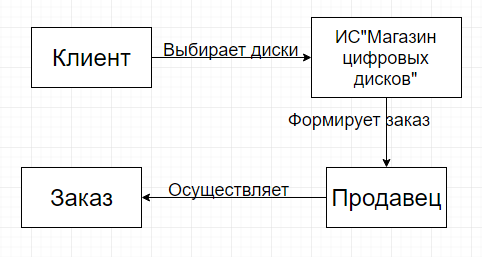
В данной курсовой работе использована такая СУБД как MySQL Server, поскольку она является наиболее оптимальным решением для поставленного вопроса. Информационная система реализована с помощью Django (Python web framework). Среда разработки - PyCharm.

1 АНАЛИЗ И КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

В данной курсовой работе будет рассмотрена информационная система «Интернет-магазин электроники».

Данная система предназначена для просмотра и покупки пользователями электронных товаров. Система позволит продавцам вести учет товаров, собирать статистику о товарах, которые продаются лучше всего. Покупатели смогут просмотреть всю необходимую информацию о них. Также пользователи смогут выбрать покупки на свой вкус, не имея заранее четкого плана.

Функциональная структура схематически представлена на рисунке 1.1. Данная структура описывает основные взаимодействия, которые происходят в процессе просмотра товаров. Так как данная система будет размещена непосредственно в магазине, и будет служить лишь инструментом поиска для пользователя - то регистрироваться ему нет смысла. Администратору также нет необходимости регистрироваться, так как данные для входа будут уже заложены в системе. Администратор - это продавец, так как в таких магазинах редко бывает много персонала и зачастую владелец и есть продавец, или же продавец занят всеми управленческими делами. Поэтому не стоит разделять продавца и администратора, для данной информационной системы. Администратор может редактировать информацию о дисках. Также он будет добавлять новые диски, которые купит у поставщика по накладной, которую сможет оформить в данной ИС, или же закажет отдельный диск, что является не такой уж редкостью, но это не стоит реализовывать, так как это скорее исключение из правил, нежели частое действие (зачастую диски поставляются партиями и лишь иногда можно попросить продавца достать некий редкий диск). Также продавец сможет получить статистику покупок, дабы добавить скидки редко покупаемые диски. Также нужно иметь алгоритм, который будет печатать чеки по заранее заготовленному шаблону. Возможности доступные пользователю админ тоже будет иметь.

  
Рисунок 1.1 – Функциональная структура информационной системы «Магазина цифровых компакт-дисков»

Информационные потребности пользователей данной информационной системы следующие:

Для пользователей:

а) Поиск товаров по названию, типу, множеству атрибутов, интервалам числовых атрибутов;

б) сортировка товаров по цене или рейтингу;

в) добавление товара в корзину;

г) добавление товара в избранное;

д) оценка товара;

Для администратора:

а) добавление новых товаров в базу;

б) редактирование информации в базе данных;

в) получение отчета по недавним продажам;

г) просмотр статистики о стоимости проданных товаров за последний 12 месяцев;

д) просмотр статистики о суммарной цене проданных товаров каждого производителя для каждой категории товаров за последний месяц;

е) оформление покупок;

Описание объектов ПО и их атрибутов:

а) Товар: Id товара, цена товара, название товара, описание товара, id подкатегории, id титульного изображения;

б) Заказ: Id заказа, адрес доставки, id состояния заказа, контактный номер, имя получателя;

в) Покупатель: Id покупателя, пароль, email, дата регистрации;

г) Корзина: Id корзины, Id покупателя, Id заказа;

д) Покупка: Id товара, Id корзины, кол-во товара

Алгоритмические зависимости:

а) расчет рейтинга товара как среднее по всем его оценкам;

б) рекомендованные товары исходя из недавно посещенных страниц товаров и их рейтингов;

Ограничения целостности для БД:

а) каждому диску соответствует только одно описание типа (аудио видео, игра);

б) не может существовать двух одинаковых товаров, их первичные ключи не могут совпадать;

в) каждый пользователь получает доступ только к функционалу, согласно роли;

г) покупка не может содержать продукты, которых нет в наличии;

д) покупка должна соответствовать существующему товару;

е) не может существовать двух одинаковых товаров, их первичные ключи не могут повторятся;

Для выполнения поставленных задач и целей будет создана информационная система «Интернет-магазина электроники» средствами фреймворка Django. Приложение будет разработано в среде PyCharm с использованием СУБД MySQL server.

2 ПОСТАНОВКА ЗАДАЧИ

Необходимо реализовать процесс взаимодействия покупателя с информационной системой, а также процесс взаимодействия администратора и системы.

Набор пользовательских возможностей у покупателя будет ограничиваться просмотром, оцениванием товаров и добавлением их в избранное/корзину/заказ. Для этого также необходимо реализовать систему регистрации

Набор возможностей администратора будет состоять из редактирования содержимого магазина и просмотра заказов, а также статистических данных.

Система должна выполнять следующие задачи:

а) отображение товаров по категориям;

б) выведение информации о конкретном товаре;

в) поиск дисков по названию;

г) фильтрация товаров по наличию тех или иных атрибутов, по интервалам атрибутов числового типа;

д) сортировка дисков по цене, рейтингу;

з) вывод всех когда-либо совершенных покупателем заказов;

и) возможность добавления новых товаров в базу данных;

й) возможность редактирования информации о товарах;

к) оценивание товаров пользователем;

л) добавление пользователем товаров в избранное.

3 ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

3.1 Построение Use Case диаграммы

В Use Case диаграмме, на рисунке 3.1, отображается взаимодействие клиента, продавца и информационной системы. При этом клиент не имеет доступа к функциям, изменяющим содержимое магазина.

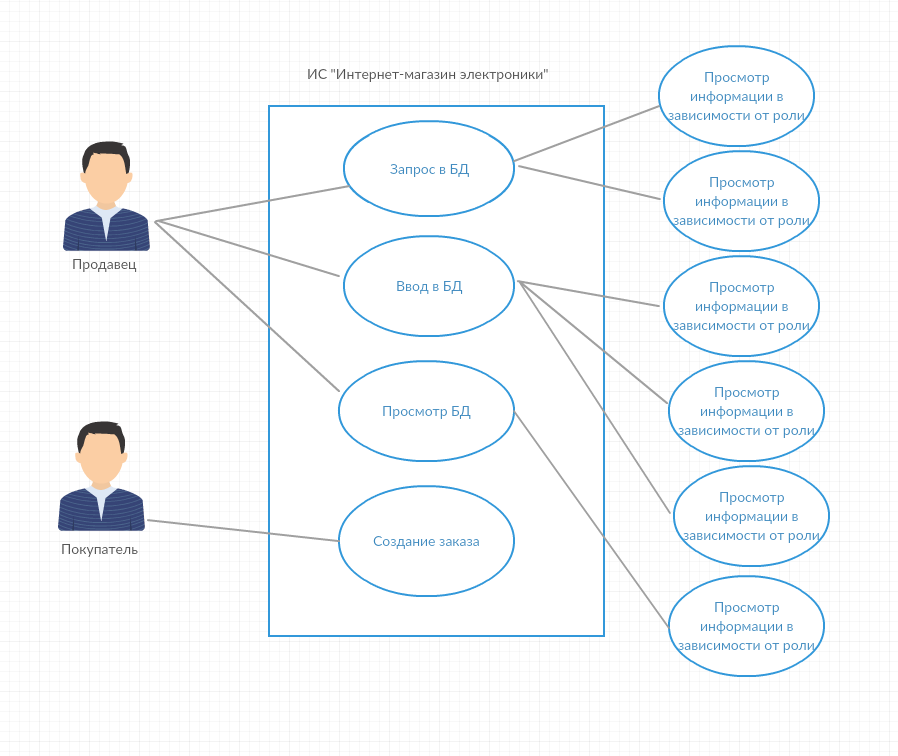


Рисунок 3.1 – Use Case диаграмма

3.2 Построение ER-диаграммы

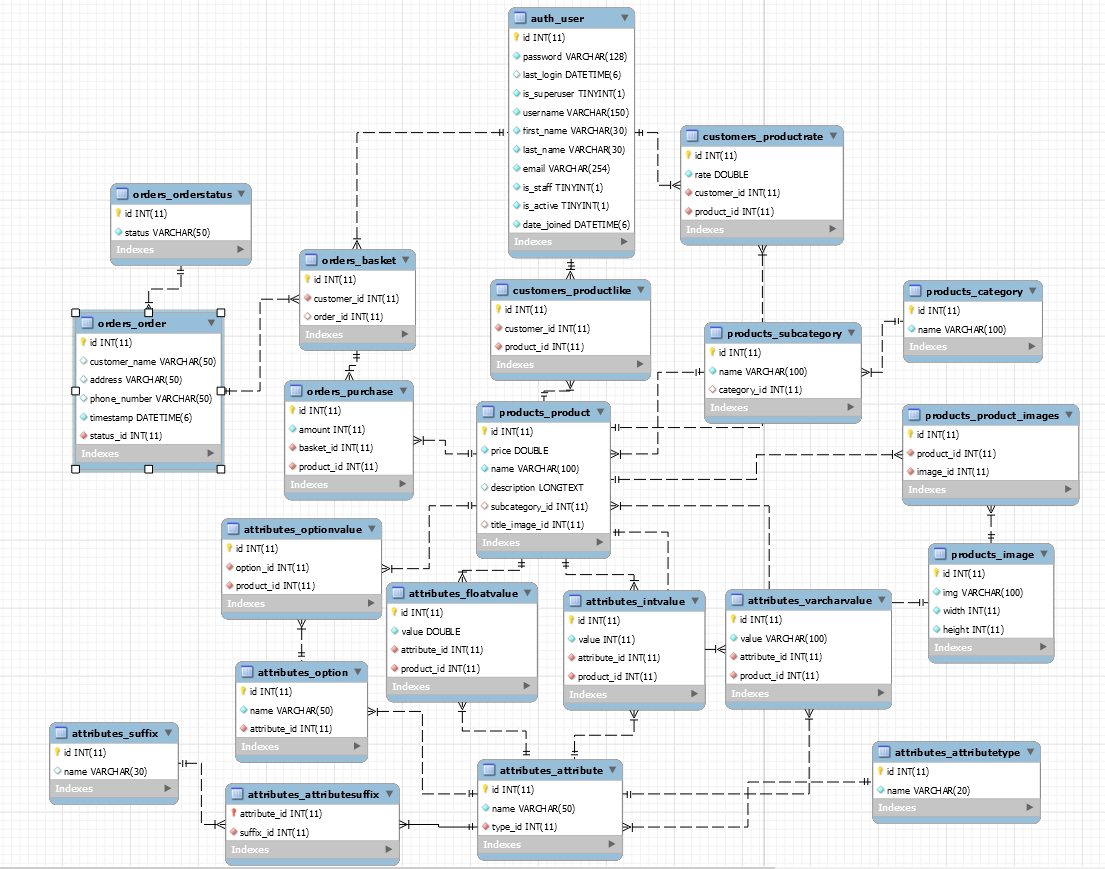


Рисунок 3.2 – ER-диаграмма

3.3 Построение схемы реляционной базы данных в третей нормальной форме:

Нормализация – разделение таблицы на две или больше с целью исключения избытка информации и избегания аномалий баз данных.

1НФ представляет собой отношение. Для него характерны атомарность значений на пересечении столбцов и строк, кортежи и атрибуты должны быть неупорядочены и кортежи не дублируются.

Для 2НФ характерны особенности 1НФ и полная функциональная зависимость неключевых атрибутов от первичного ключа отношения.

3НФ – состояние отношения, которое находится во 2НФ и в котором между атрибутами нет транзитивных зависимостей.

Переход к следующей нормальной форме возможен только тогда, когда удовлетворяются все условия предыдущей формы.

Для предметной области построим универсальное отношение, которое включает описание ранее выделенных объектов и их свойств (см. рис. 3.3).   
  


Рисунок 3.3 – Универсальное отношение

Данное универсальное отношение удовлетворяет требования 1НФ, то есть является отношением реляционных баз данных, а именно: кортежи не упорядочены, атрибуты не упорядочены, информация кортежей не дублируется, значения атрибутов атомарные. Это универсальное отношение не является 2НФ, так как атрибуты «Дата покупки», «Скидка», «Id диска», «Дата окончания скидки» не полностью зависят от первичного ключа. Продолжим процедуру нормализации.

2НФ будет состоять из двух таблиц.

Выделим таблицу Т1, рисунок 3.4, и таблицу Т2, рисунок 3.5.



Рисунок 3.4 – Таблица Т1

Таблица Т1 находится в 2НФ, так как все атрибуты имеют полную функциональную зависимость от первичного ключа «Id диска». Также присутствует транзитивная зависимость, что значит таблица не в 3 НФ.

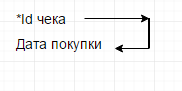


Рисунок 3.5 – Таблица Т2

Таблица Т2 находится в 2НФ и 3НФ, так как имеет всего 2 атрибута, один из которых первичный ключ. Таблица не имеет внешних ключей, что будет исправлено в процессе нормализации.

Для привидения Т1 к 3НФ разделим её на Т3, рисунок 3.6 и Т4, рисунок 3.7.

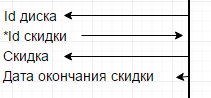


Рисунок 3.6 – Таблица Т3

Таблица Т3 находится в 3НФ, так как она находится в 2НФ и в ней отсутствуют транзитивные зависимости.



Рисунок 3.7 – Таблица Т4

Таблица Т4 находится в 3НФ, так как она находится в 2 НФ и в ней отсутствуют транзитивные зависимости.

Данные таблицы далеки от представленных в ER-диаграмме, так как хоть они и в 3НФ, но после построения связей набор атрибутов изменится, но это не отменит того, что они будут в 3НФ. Это можно увидеть на конечной схеме базы данных (в 3НФ), рисунок 3.8.

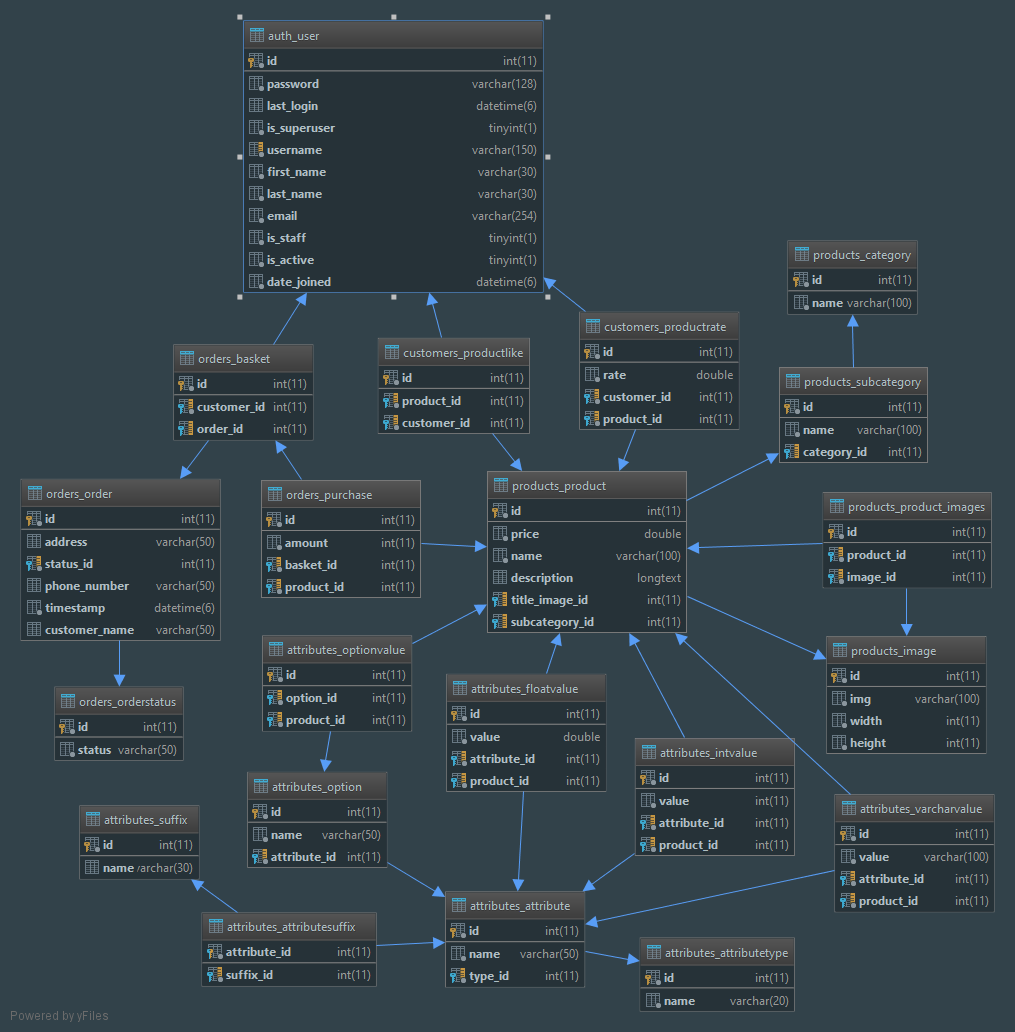


Рисунок 3.8 – Схема базы данных

4 ОПИСАНИЕ ПРОГРАММЫ

4.1 Общие сведения

Разрабатываемый программный продукт создавался в интегрированной среде разработки для языка программирования Python PyCharm. В качестве СУБД была выбрана MySQL Server. Ее основные преимуществами, безусловно, являются простота в использовании (упрощенное создание таблиц, отстутствие сложных операций управления базой данных, и т.д.), а также то, что она является одной из самых надежных платформ базы данных. Эта СУБД очень популярна, а значит она хорошо развита и содержит минимум ошибок.

4.2 Вызов и загрузка

Для установки сервера необходим django 1.7 и python 3.5.2. Сервер запускается командой python manage.py runserver и доступен через порт 8000 по умолчанию.

4.3 Назначение и логическая структура

В постановке задачи, описанной выше, указаны все пункты, которые являются обязательными для выполнения. Данная программа соответствует этим требованиям целиком и полностью.

Главных одна два, как и варианта использования программы. В окне пользователя доступны функции поиска товара по различным критериям, сортировка, добавления товара в заказ.

В окне администратора более обширный функционал. Здесь возможен просмотр, редактирование, добавление дисков и скидок, просмотр статистики покупок, оформление покупки, запись в базу с XML файла.

Выбор стиля интерфейса для пользователя обусловлен тем что пользовательский вариант системы будет запущен на сенсорном экране, а значит минимум маленьких деталей (чек-боксов, радио кнопок и т.д.).

4.4 Описание физической модели БД

В качестве СУБД была выбрана MySQL Server, поэтому описание физической модели представлено в виде скриншотов из этой программы с описанием.

Структура таблицы «DiskGoods» приведена на рисунке 4.1.

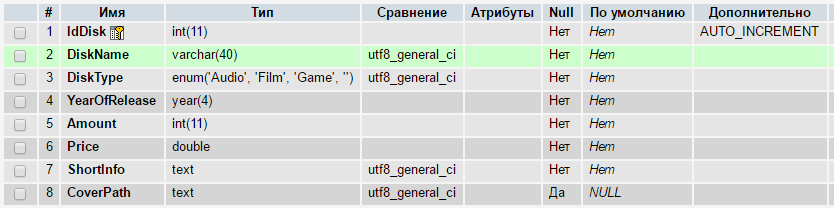


Рисунок 4.1 – Структура таблицы «DiskGoods»

На приведенном рисунке перечислены поля таблицы «DiskGoods». В качестве первичного ключа обозначено поле «IdDisk», на которое наложено ограничение непустое и уникальное. Остальные поля также не допускают значение NULL, кроме поля «CoverPath», так как не обязательно диск будет содержать обложку (будет использована обложка по умолчанию).

Структура таблицы «DiskAudio» приведена на рисунке 4.2.



Рисунок 4.2 – Структура таблицы «DiskAudio»

На приведенном рисунке перечислены поля таблицы «DiskGoods». В качестве первичного ключа обозначено поле «IdDisk», на которое наложено ограничение непустое и уникальное. Остальные поля также не допускают значение NULL.

Структура таблицы «DiscFilm» приведена на рисунке 4.3.



Рисунок 4.3 – Структура таблицы «DiskFilm»

На приведенном рисунке перечислены поля таблицы «DiskFilm». В качестве первичного ключа обозначено поле «IdDisk», на которое наложено ограничение непустое и уникальное. Остальные поля также не допускают значение NULL.

Структура таблицы «DiskGame» приведена на рисунке 4.4.



Рисунок 4.4 – Структура таблицы «DiskGame»

На приведенном рисунке перечислены поля таблицы «DiskGame». В качестве первичного ключа обозначено поле «IdDisk», на которое наложено ограничение непустое и уникальное. Остальные поля также не допускают значение NULL.

Структура таблицы «Discounts» приведена на рисунке 4.5.



Рисунок 4.5 – Структура таблицы «Discounts»

На приведенном рисунке перечислены поля таблицы «Discounts». В качестве первичного ключа обозначено поле «IdDiscount», на которое наложено ограничение непустое и уникальное. Остальные поля также не допускают значение NULL.

Структура таблицы «Checks» приведена на рисунке 4.6.

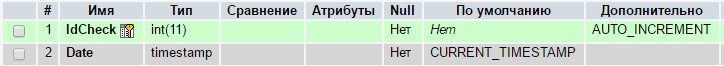


Рисунок 4.7 – Структура таблицы «Checks»

На приведенном рисунке перечислены поля таблицы «Checks». В качестве первичного ключа обозначено поле «IdDiscount», на которое наложено ограничение непустое и уникальное. Поле «Date» также не допускают значение NULL и имеет значение по умолчанию (текущую дату).

Структура промежуточной таблицы «Purchase» приведена на рисунке 4.7.

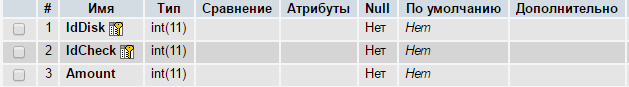


Рисунок 4.7 – Структура промежуточной таблицы «Purchase»

На приведенном рисунке перечислены поля промежуточной таблицы «Checks» (между «DiskGoods» и «Checks»). В качестве составного первичного ключа обозначено два поля «IdCheck» и «IdDisk», на которые наложено ограничение непустое и уникальное. Поле «Amount» также не допускают значение NULL.

4.5 Описание программной реализации

На рисунке 4.8 приведено главное меню пользователя.



Рисунок 4.8 – Главное меню пользователя

Это основное окно взаимодействия для пользователя. Поиск осуществляется по имени совпадению с именами дисков (строка, введенная в поле поиска) по типу, который определится нажатием на нужную кнопку типа. Также присутствует расширенный поиск, который реализован в виде всплывающего окна, приведенного на рисунке 4.9.

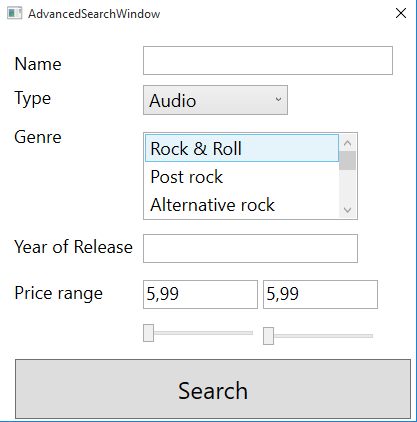


Рисунок 4.9 – Окно расширенного поиска

В данном окне можно указать более широкий спектр критериев поиска, чем в основном меню. При этом неизменные значения берутся из базы, дабы сузить круг возможных вариантов (жанр, в зависимости от типа и минимальная/максимальная цена диска).

При клике на определенный диск всплывет окно подробной информации, в котором указаны остальные поля диска, не выведенные на основном меню (на основном только краткая информация) приведенное на рисунке 4.10:

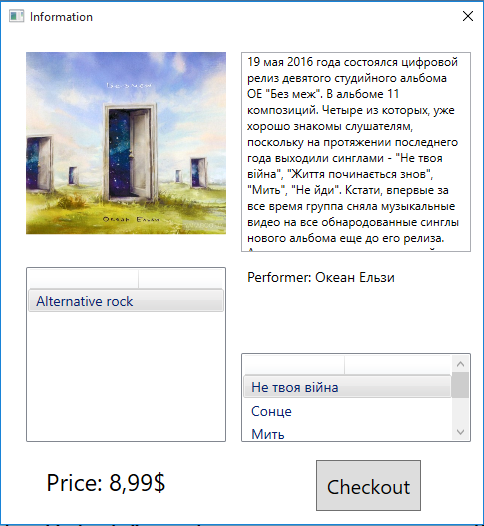


Рисунок 4.11 – Окно подробной информации

При нажатии на кнопку «Checkout» произойдет добавление диска в заказ, который затем сможет оформить продавец, из меню продавца, что и представлено на рисунке 4.12.

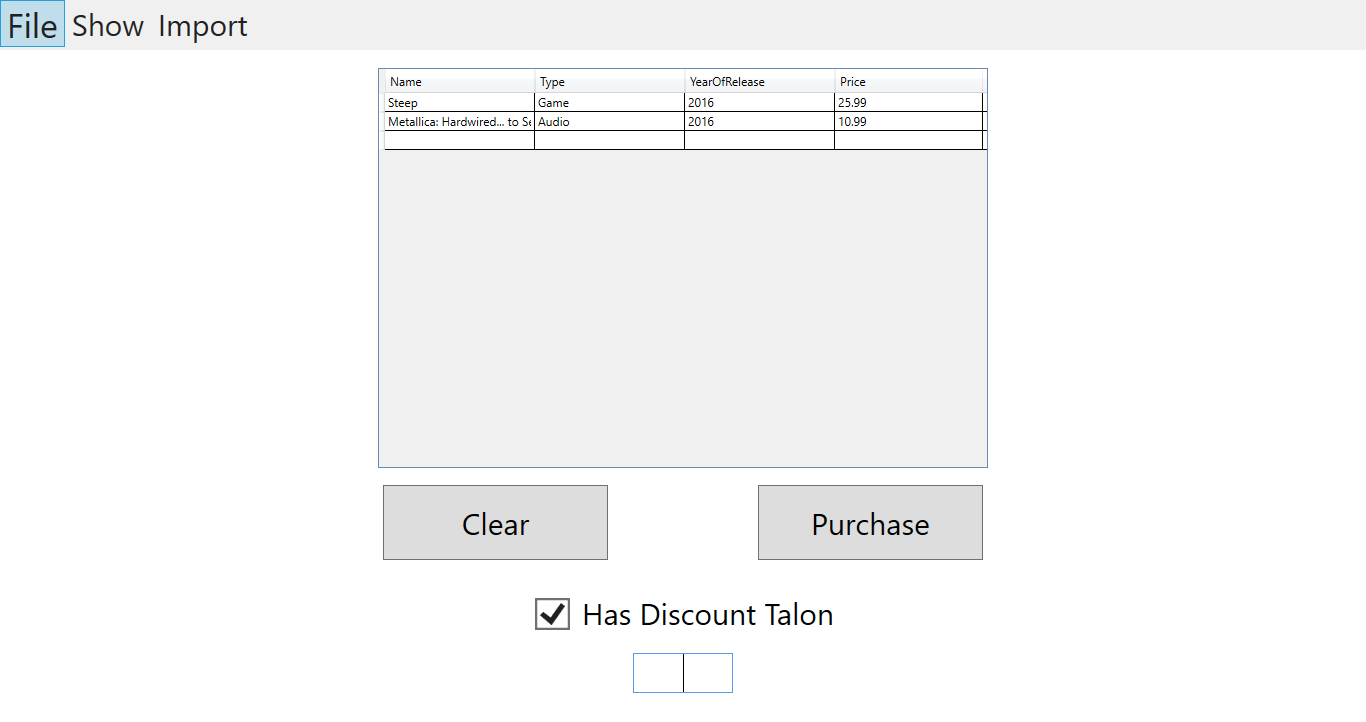


Рисунок 4.12 – Главное меню продавца

При нажатии на клавишу «Purchase» будет произведена покупка, список которой отображен в списке в центре меню. При нажатии на кнопку «Clear» список очистится. Также есть чекбокс, который позволяет указать скидку на покупку, если пользователь предоставит дисконтную карту, заверенную подписью продавца. Пример текста (подразумевается, что карта будет напечатана на спец установке, как и кассовый чек) такой карты приведен на рисунке 4.13.

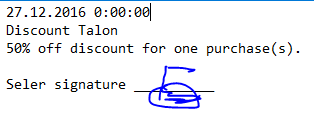


Рисунок 4.13 – Дисконтная карта

На карте указан процент скидки, дата выдачи, и место для подписи продавца, для её заверения.

При совершении покупки покупатель получит кассовый чек, пример которого приведен на рисунке 4.14.

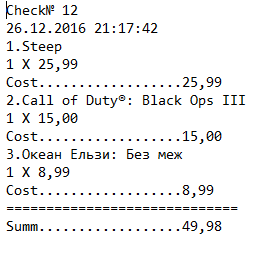


Рисунок 4.14 – Фискальный кассовый чек

В верхней части меню продавца представлена строка меню, она позволяет перейти к окнам: окно дисков (см. рисунок 4.15), окно скидок (см. рисунок 4.17), окно статистик (см рисунок 4.20). Также в меню есть пункты для закрытия программы, и импорта в базу из XML файла.

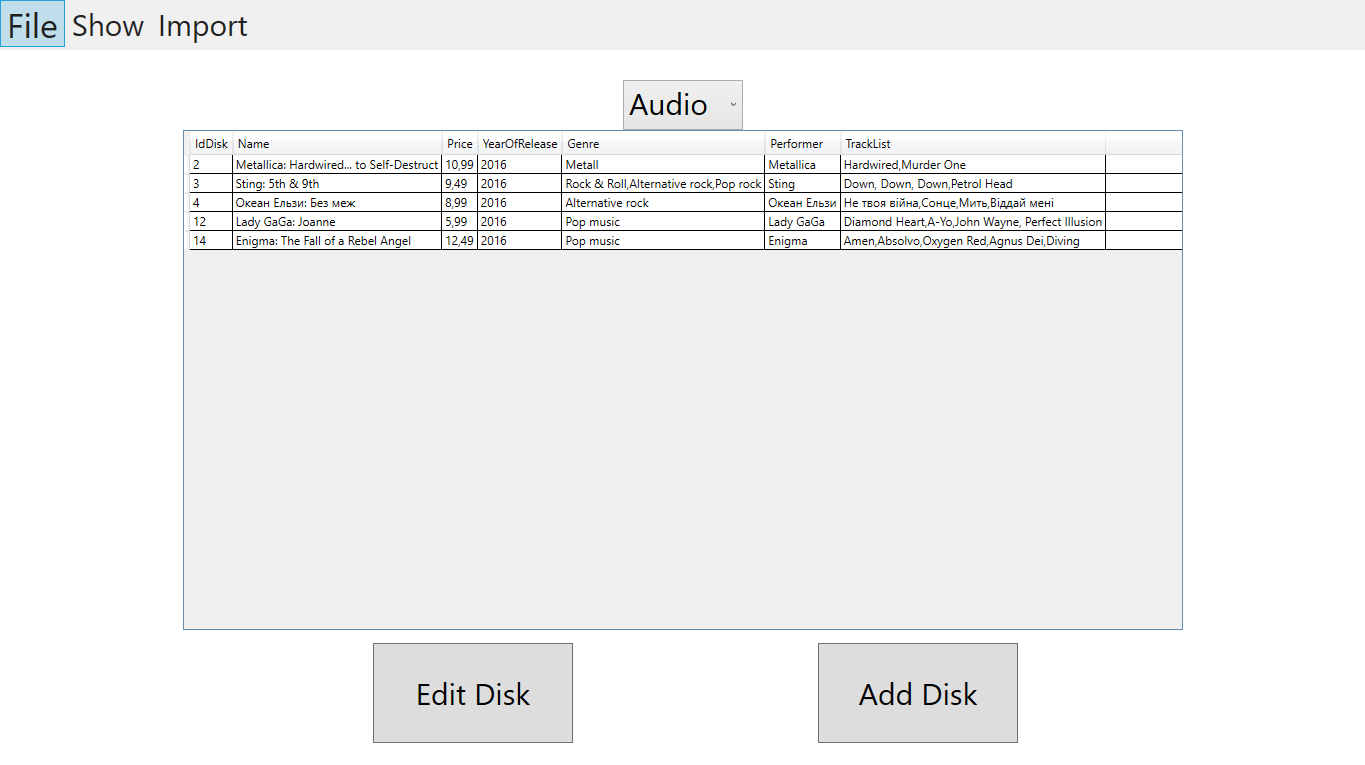


Рисунок 4.15 – Окно дисков

В данном окне отображаются диски, в зависимости от типа, выбранного в комбобоксе над списком. По нажатию на кнопку «Add Disk» откроется окно добавления нового диска, приведенное на рисунке 4.16.

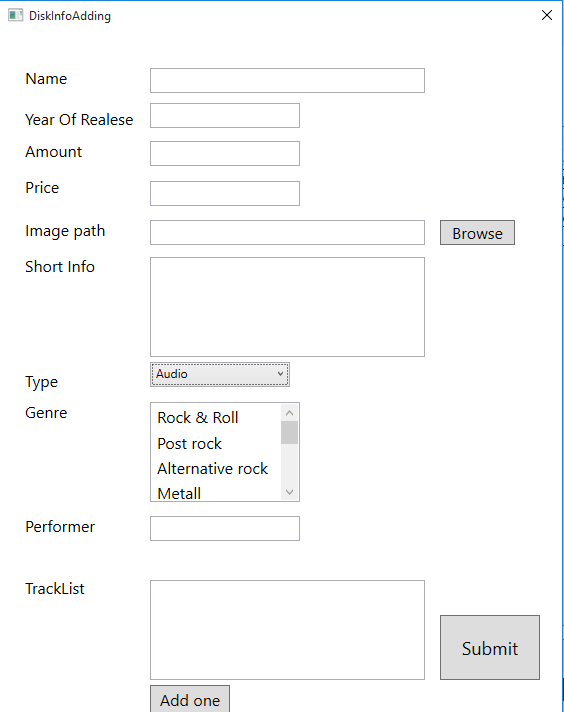


Рисунок 4.16 – Окно добавления нового диска

Окно редактированием диска аналогично с окном добавления. Разница в том, что при редактировании поля уже содержат значения. Окно меняет свои поля в зависимости от выбранного типа, так как в разных типах диска различные поля. При редактировании тип изменить нельзя.

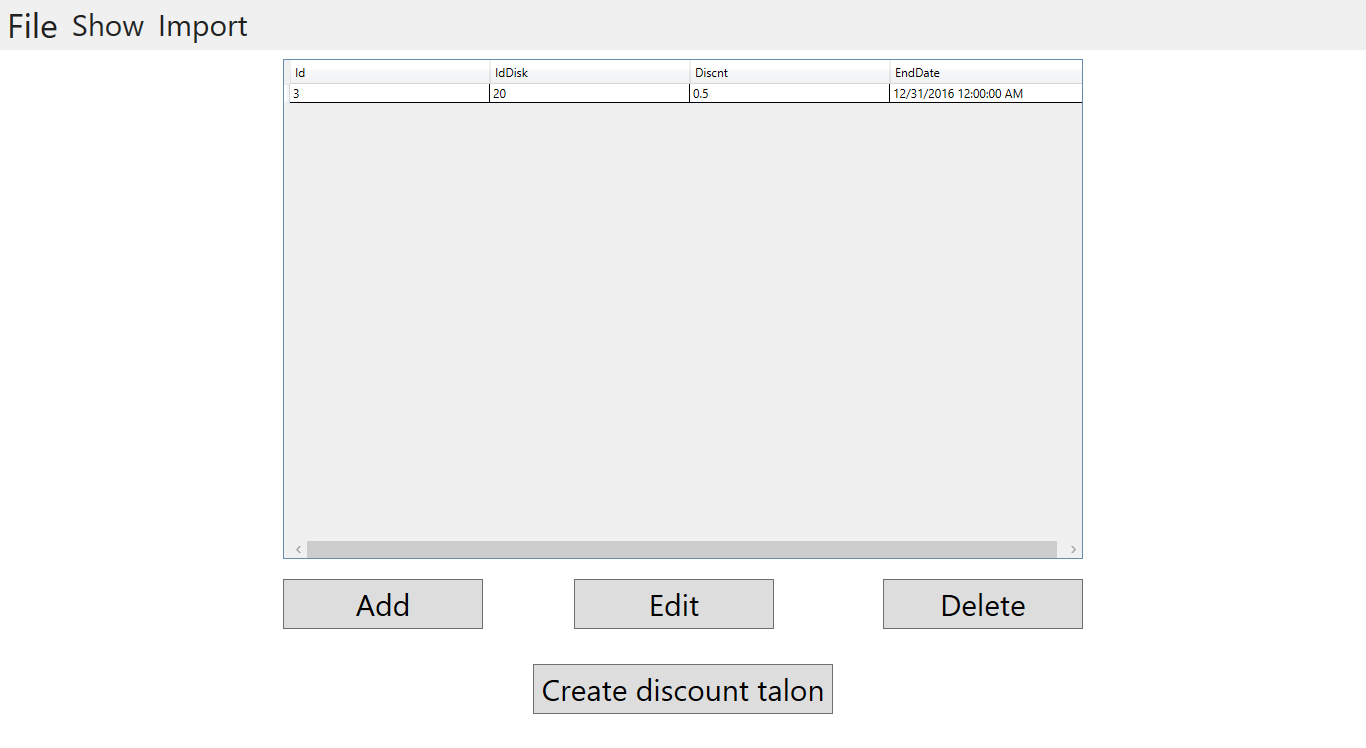


Рисунок 4.17 – Окно скидок

В окне предоставлен список всех скидок. Также присутствует возможность добавления, редактирования и удаления скидки соответствующими кнопками. Окно добавления скидки приведено на рисунке 4.18.

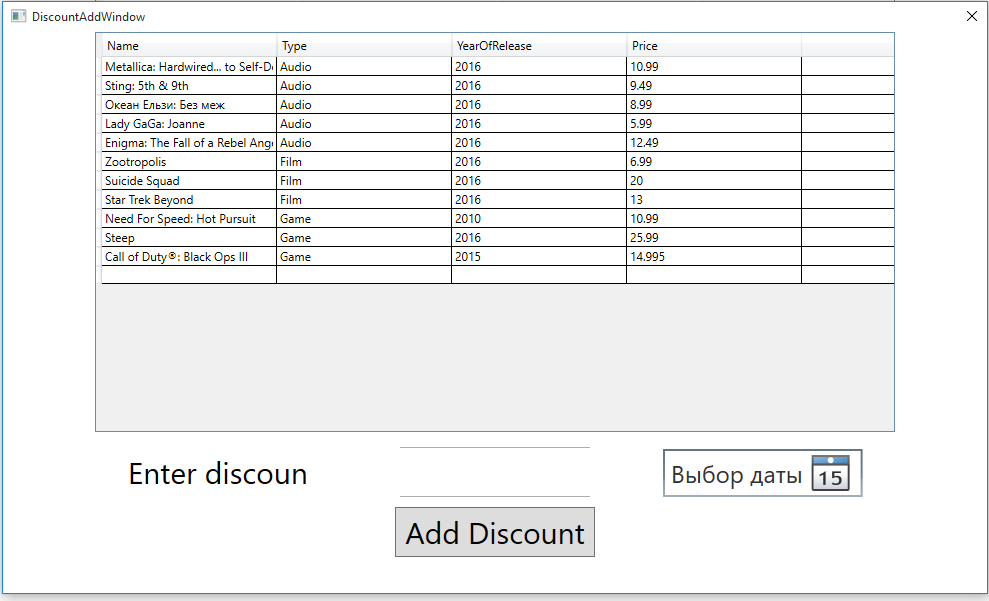


Рисунок 4.18 – Окно добавления скидки

В данном окне нужно выбрать диск из списка всех дисков, затем ввести в поле для ввода скидку, выбрать дату её окончания и нажать на кнопку «Add Discount». Это всплывающее окно.

На рисунке 4.19 приведено диалоговое окно по созданию дисконтной карты:

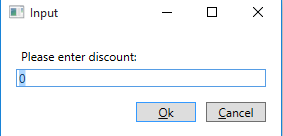


Рисунок 4.19 – Диалоговое окно создания дисконтной карты

Формат карты приведен выше (см. рисунок 4.13).

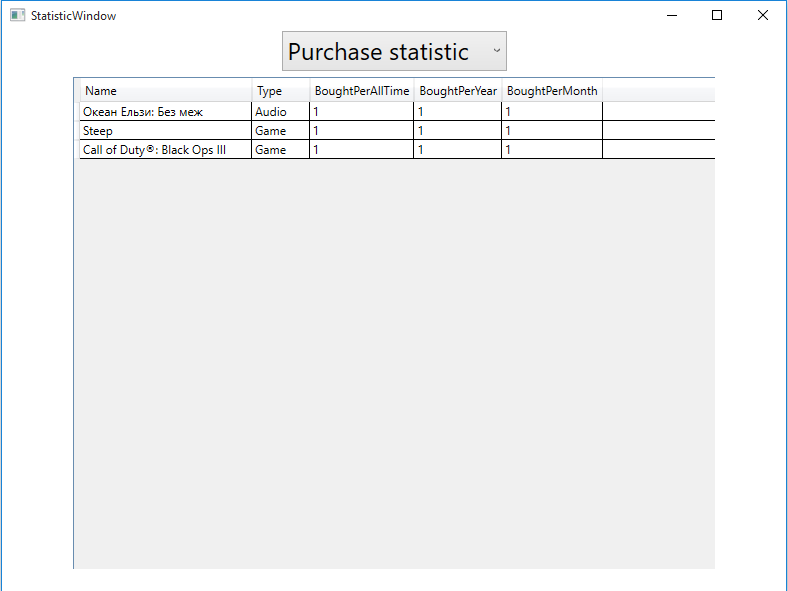


Рисунок 4.20 – Окно статистики

На данном окне показана статистика покупок всех дисков за все время, за год и за месяц (год и месяц берется текущий календарный). Также при выборе в комбобоксе можно просмотреть топ покупаемых дисков, рисунок 4.21.

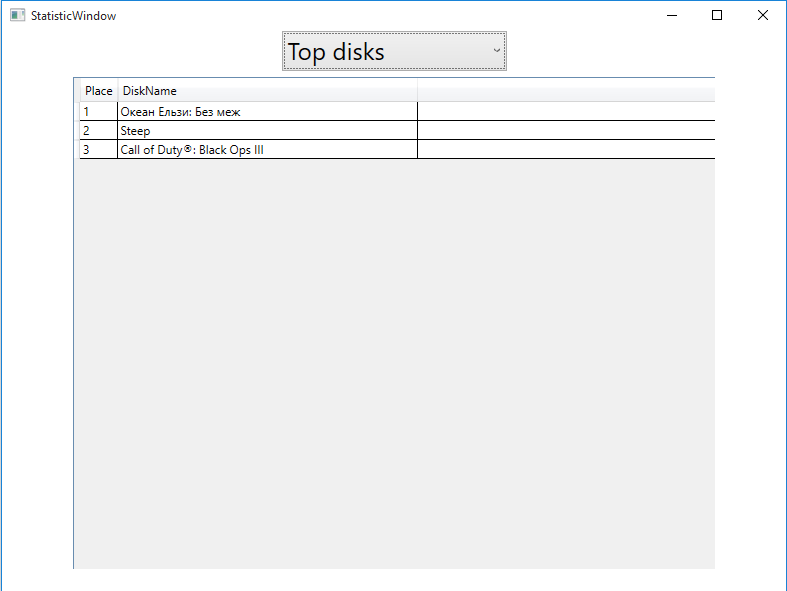


Рисунок 4.21 – Статистика: топ дисков

Далее приведены примеры запросов, использованных для получения информации.

Расширенный поиск среди товаров:

"SELECT \* FROM `DiskGoods`, `DiskAudio` WHERE `DiskGoods`.`IdDisk` = `DiskAudio`.`IdDisk` AND LOWER(`DiskName`) LIKE('%" + searchText.ToLower() + "%') AND `DiskType` = '" + type + "' AND `YearOfRelease` = " + parametr[0] + "AND `Genre` LIKE ('%" + parametr[3] + "%') AND `Price` BETWEEN " + parametr[1].ToString().Replace(",", ".") + " AND " + parametr[2].ToString().Replace(",", ".") + ";"

Получение скидки на товар:

$@"SELECT `Discount`, `EndDate` FROM `Discounts` WHERE `IdDisk` = {disk.Id} AND `EndDate` >= DATE(NOW()) ;"

Получение возможных жанров, для заданного типа дисков:

$@"SELECT column\_type FROM information\_schema.columns WHERE table\_name = 'Disk{type}' AND column\_name = 'Genre';"

Получение статистики покупок за текущий месяц:

$@"SELECT `DiskGoods`.\* , SUM(`Purchase`.`Amount`) AS 'Statistic' FROM `DiskGoods`, `Purchase` , `Checks` WHERE MONTH(`Checks`.`Date`) = MONTH(CURDATE()) AND YEAR(`Checks`.`Date`) = YEAR(CURDATE()) AND `DiskGoods`.`IdDisk` = `Purchase`.`IdDisk`AND Purchase.IdCheck = Checks.IdCheck GROUP BY `Purchase`.`IdDisk`;"

Поиск без дополнительных параметров:

"SELECT \* FROM `DiskGoods`, `DiskFilm` WHERE `DiskGoods`.`IdDisk` = `DiskFilm`.`IdDisk` AND LOWER(`DiskName`) LIKE('%" + searchText.ToLower() + "%') AND `DiskType` = '" + type + "';"

Метод импорта из внешнего источника (задача автоматизации):

public static List<Model.Disk> Xml(string path) {

List<Model.Disk> xmlResultList = new List<Model.Disk>();

//задаем путь к нашему рабочему файлу XML

string fileName = @path;

//читаем данные из файла

XDocument doc = XDocument.Load(fileName);

//проходим по каждому элементу в найшей library

//(этот элемент сразу доступен через свойство doc.Root)

foreach (XElement el in doc.Root.Elements())

{

//Выводим имя элемента и значение аттрибута id

xmlResultList.Add(new Model.Disk(0,

el.Attribute("name").Value,

"Audio",

Convert.ToInt32(el.Attribute("year").Value),

Convert.ToInt32(el.Attribute("amount").Value),

Convert.ToDouble(el.Attribute("price").Value , CultureInfo.InvariantCulture),

el.Element("info").Value,

el.Element("path").Value,

el.Element("genre").Value.Split(',').ToList(),

el.Element("performer").Value,

el.Element("trackList").Value.Split('\*').ToList()));

DBControl.Query($@"INSERT INTO `DiskGoods`(`IdDisk`, `DiskName`, `DiskType`, `YearOfRelease`, `Amount`, `Price`, `ShortInfo`, `CoverPath`) VALUES (NULL,'{ el.Attribute("name").Value}','{"Audio"}',{el.Attribute("year").Value},{el.Attribute("amount").Value},'{el.Attribute("price").Value}','{el.Element("info").Value}','{el.Element("path").Value}');");

DBControl.Query($@"INSERT INTO `Shop`.`DiskAudio` (`IdDisk`, `Genre`, `Performer`, `TList`) VALUES({GetMaxIndex()},'{el.Element("genre").Value}','{el.Element("performer").Value}','{el.Element("trackList").Value}');");

}

return xmlResultList;

}

ВЫВОДЫ

В данной курсовой работе была разработана информационная система «Магазин цифровых компакт-дисков». Она была создана в форме WPF приложения на платформе .NET с использованием паттерна MVC.

Данная система может использоваться в магазине компакт-дисков, хоть и как прототип, поскольку главной задачей было реализовать программу максимально подходящую под данную предметную область.

Работа над информационной системой требовала детального разбора будущей базы данных, в ходе которого были изучены различные этапы оптимизации базы данных и проекта в целом. Среди них – ER-диаграмма, нормализация, концептуальный анализ, и т.д.

Все это в совокупности позволило получить качественную и работоспособную информационную систему, что удовлетворяет все требования, которые были поставлены. Наиболее ценным в этом проекте являются приобретенные навыки работы с различными СУБД и закрепление знаний в направлении .NET. Кроме того, использование SQL-запросов, и т.д. несомненно способствуют разностороннему и гармоничному развитию студента как программиста.

ПЕРЕЧЕНЬ ССЫЛОК

1. Дюбуа П. MySQL Сборник рецептов[Текст] / П. Дюбуа. – Санкт-Петербург: Символ-Плюс. – 1056 с. – (Вне серии).
2. Тахагхогхи С. Руководство по MySQL[Текст] / С. Тахагхогхи, Х. Вильямс. – Москва: Русская редакция, 2007. – 544 с. – (Бестселлеры O`Reilly).
3. Гольцман В. И. MySQL 5.0. Библиотека программиста[Текст] / В. И. Гольцман., 2010. – 253 с. – (Библиотека программиста).